Stochastic Model Predictive Control with Optimal Linear Feedback for Mobile Robots in Dynamic Environments

Yunfan Gao [*,**]**, Florian Messerer** [**]**, Niels van Duijkeren** [*]**,**
Moritz Diehl [**]

* Robert Bosch GmbH, Corporate Research, Stuttgart, Germany
(e-mail: {yunfan.gao, niels.vanduijkeren}@de.bosch.com).
** Department of Microsystems Engineering (IMTEK), University of
Freiburg, Germany (e-mail: {florian.messerer,
moritz.diehl}@imtek.uni-freiburg.de)

Abstract: Robot navigation around humans can be a challenging problem since human movements are hard to predict. Stochastic model predictive control (MPC) can account for such uncertainties and approximately bound the probability of a collision to take place. In this paper, to counteract the rapidly growing human motion uncertainty over time, we incorporate state feedback in the stochastic MPC. This allows the robot to more closely track reference trajectories. To this end the feedback policy is left as a degree of freedom in the optimal control problem. The stochastic MPC with feedback is validated in simulation experiments and is compared against nominal MPC and stochastic MPC without feedback. The added computation time can be limited by reducing the number of additional variables for the feedback law with a small compromise in control performance.

Keywords: robotics, stochastic model predictive control, motion planning

1. INTRODUCTION

Increasingly many autonomous mobile robots (AMRs) operate alongside humans in industry. A fundamental prerequisite for AMRs to work in a shared space with humans is that they do not collide with humans (Brock and Khatib, 2002). In order to coordinate robot motions effectively, it is essential to consider predictions of human trajectories, which are generally uncertain. Uncertainty-aware planning often includes probabilistic constraints for collision avoidance (Du Toit and Burdick, 2012; Carvalho et al., 2014; Zhu and Alonso-Mora, 2019). A real-time capable approach for dynamic obstacle avoidance using stochastic model predictive control (MPC) is presented in (Castillo-Lopez et al., 2020).

Human motion uncertainty typically grows rapidly over time. Large uncertainties cause the robot to keep an overly big distance to the humans, potentially deviating from intended reference trajectories. Incorporating a feedback law in the optimal control problem (OCP) formulation can mitigate conservativeness (Scokaert and Mayne, 1998). The scheme can be crudely approximated by keeping the covariance matrices constant after some time steps in the future (Yan and Bitmead, 2005) or bounding the matrices by pre-defined values (Schwarting et al., 2018). These approaches, however, overlook the resulting uncertainty of robot state and the discrepancy in the robot feedback capacities at different robot states.

In robust and stochastic MPC, several research works explicitly model the feedback policies parameterized by, e.g., linear feedback laws (Nagy and Braatz, 2004). The associated feedback gain matrices can either be precomputed (Mayne et al., 2011) or optimized online (Messerer and Diehl, 2021). For motion planning of nonhonolomic robot systems, e.g., differential-drive or tricycle robots, precomputed constant feedback gains are unlikely to be effective for all combinations of robot states and human positions. Therefore, treating the gain matrices as decision variables is especially beneficial. Meanwhile, it results in optimization problems that are in general challenging to solve (Goulart et al., 2006). Incorporating feedback on human motion uncertainty in motion planning with the feedback gain matrices being optimized has not yet been extensively investigated in the existing literature.

In this paper, we present a stochastic MPC formulation for robot motion planning; focusing on Gaussian-distributed human motion uncertainty and incorporating feedback on the uncertainty. By probabilistically avoiding collisions with people in the MPC prediction horizon and enforcing the robot to almost stand still after that, we aim for a concept called *passive safety* (Macek et al., 2008). While the feedback can reduce the uncertainty of the human-robot distance, it increases the uncertainty of the robot state. Therefore, it is desired that feedback is weaker when the human is further away. We accomplish this through optimizing a time-varying linear feedback law in the OCP, coupled with carefully designed cost terms and constraints in the OCP formulation.

We discuss and compare the usage of two feedback policies and a baseline without feedback. One feedback policy considers both the robot and human state uncertainty while the other considers only the human. The performance and the solution time are presented, highlighting the possibility to trade off computation time with control performance.

Notation The nominal system state is denoted by x and the disturbed system state is denoted by $\tilde{x}$. Let $\bar{x}$ be the currently measured system state. An all-ones vector of the size n is written as $\mathbb{1}_n$. The notation of a zero matrix of the size $m \times n$ is $[0]_{m \times n}$. A set of natural numbers for the interval $[a, b]$, possibly starting from 0, is denoted by $\mathbb{N}_{[a,b]}$. The quadratic norm weighted by a symmetric positive definite matrix Q is denoted by $\|x\|_Q^2 := x^\top Q x$. A multi-variable Gaussian distribution with mean μ and variance Σ is represented by $\mathcal{N}(\mu, \Sigma)$.

2. MODEL PREDICTIVE CONTROL PROBLEM

Robot Model Consider a kinematic model of a differential-drive robot with center position (p_x^r, p_y^r) and heading θ (Siegwart et al., 2011). The forward and angular velocities are denoted by v^r and ω, respectively:

$$x^r := \begin{bmatrix} p_x^r & p_y^r & \theta & v^r & \omega \end{bmatrix}^\top \in \mathbb{R}^5. \tag{1}$$

The control input u^r consists of forward acceleration a^r and angular acceleration α, i.e., $u^r := \begin{bmatrix} a^r & \alpha \end{bmatrix}^\top \in \mathbb{R}^2$. The continuous-time equations of motion are

$$\dot{x}^r = \begin{bmatrix} v^r \cos(\theta) & v^r \sin(\theta) & \omega & a^r & \alpha \end{bmatrix}^\top. \tag{2}$$

Consider a zero-order-hold control signal with discretization intervals of length δ_t. The system is discretized by numerical integration (explicit Runge-Kutta integrator of order four):

$$x_{k+1}^r = \psi_k^r(x_k^r, u_k^r) \colon \mathbb{R}^5 \times \mathbb{R}^2 \to \mathbb{R}^5. \tag{3}$$

The robot is subject to affine constraints on v^r, ω, a^r, and α due to actuator limits. The nonlinear collision avoidance constraints will be detailed in a later paragraph.

Human Model The human model, as we will see in Section 3.1, plays a role in the uncertainty propagation of the system state when the control policy incorporates feedback on human motion uncertainty. The state of the human is its center position and its input is the velocity:

$$x^h := \begin{bmatrix} p_x^h & p_y^h \end{bmatrix}^\top, \ u^h := \begin{bmatrix} v_x^h & v_y^h \end{bmatrix}^\top, \ \text{and} \ \dot{x}^h = u^h. \tag{4}$$

The input of the human motion $\tilde{u}^h(t)$ is also modeled as a zero-order hold signal with the same step length δ_t. The input $\tilde{u}_k^h$ is assumed to be Gaussian-distributed:

$$\tilde{u}_k^h \sim \mathcal{N}(u_k^h, W_k^h), \ \text{for} \ k \in \mathbb{N}_{[0,N-1]},$$

where mean u_k^h and covariance $W_k^h \in \mathbb{R}^{2 \times 2}$ are time-varying. The discrete-time system dynamics are

$$\tilde{x}_{k+1}^h = A^h \tilde{x}_k^h + B^h \tilde{u}_k^h, \ \text{for} \ k \in \mathbb{N}_{[0,N-1]}. \tag{5}$$

We assume that the current human state is measured without uncertainty. The human does not react to the robot movement. The human velocity prediction and covariance estimates across any two subsequent predictions are consistent.

Collision Avoidance Denote the distance between the robot position and the human position as

$$\Delta_{\mathrm{rh}}(x^r, x^h) := \left\| \begin{bmatrix} p_x^r & p_y^r \end{bmatrix}^\top - \begin{bmatrix} p_x^h & p_y^h \end{bmatrix}^\top \right\|_2.$$

The distance shall not be less than a minimum collision-free distance Δ_{safe}, i.e.,

$$\Delta_{\mathrm{rh}}(x^r, x^h) - \Delta_{\mathrm{safe}} \geq 0. \tag{6}$$

Terminal Constraints We impose a constraint requiring the robot to be nearly stationary at the terminal state:

$$0 \leq v_N^r \leq \epsilon_v. \tag{7}$$

The upper bound is a small positive constant ϵ_v. The control policy beyond the terminal state is to remain standing still. The terminal constraints together with the stabilizing control policy ensure that the robot is passively safe, i.e., the robot does not collide while moving. Here, the robot is allowed to have a non-zero terminal angular velocity due to its circular shape.

Reference Trajectory Tracking We assume that state and input reference trajectories are given:

$$\left(x_0^{r,\mathrm{ref}}, \dots, x_N^{r,\mathrm{ref}}\right) \ \text{and} \ \left(u_0^{r,\mathrm{ref}}, \dots, u_{N-1}^{r,\mathrm{ref}}\right). \tag{8}$$

The stage and terminal costs of the OCP are weighted squared tracking errors:

$$l_k(x_k^r, u_k^r) := \frac{1}{2}\|u_k - u_k^{r,\mathrm{ref}}\|_R^2 + \frac{1}{2}\|x_k - x_k^{r,\mathrm{ref}}\|_Q^2, \tag{9a}$$

$$l_N(x_N^r) := \frac{1}{2}\|x_N^r - x_N^{r,\mathrm{ref}}\|_{Q_e}^2. \tag{9b}$$

Problem 1. Given predictions of a human motion trajectory and human motion uncertainty, the aim is to find a control policy incorporating *feedback* on human motion uncertainty such that the robot tracks a reference trajectory (8) while probabilistically satisfying the input-state constraints and collision avoidance constraints (6).

3. STOCHASTIC OCP WITH OPTIMIZED LINEAR STATE FEEDBACK

The system state $x \in \mathbb{R}^7$ consists of the concatenation of the robot and the human states. The control input in the OCP is the actual input u^r of the robot. The input of the human motion model is considered as a stochastic disturbance. The control policy incorporates feedback on the state deviation $(\tilde{x}_k - x_k)$. The feedback scheme is parameterized as a time-varying linear feedback policy:

$$\tilde{u}_k^r(\tilde{x}_k) := u_k^r + K_k(\tilde{x}_k - x_k), \tag{10}$$

with $K_k \in \mathbb{R}^{2 \times 7}$ for $k \in \mathbb{N}_{[1,N-1]}$. For $k = 0$, we do not consider a feedback gain because the current human state is assumed to be known exactly.

3.1 Covariance Propagation

The state covariance matrices Σ_k consist of the robot state covariance $\Sigma_k^r \in \mathbb{R}^{5 \times 5}$ and the human state covariance $\Sigma_k^h \in \mathbb{R}^{2 \times 2}$ as diagonal blocks. The off-diagonal blocks $\Sigma_k^{rh} \in \mathbb{R}^{5 \times 2}$ represent the covariance between the robot and human states:

$$\Sigma_k := \begin{bmatrix} \Sigma_k^r & \Sigma_k^{rh} \\ \{\Sigma_k^{rh}\}^\top & \Sigma_k^h \end{bmatrix}.$$

The covariance propagation can be approximated based on the linearization of the discretized system dynamics at the nominal trajectory. Let

$$A_k^r := \frac{\partial \psi_k^r}{\partial x_k^r}(x_k^r, u_k^r), \quad B_k^r := \frac{\partial \psi_k^r}{\partial u_k^r}(x_k^r, u_k^r), \tag{11a}$$

$$\check{A}_k := \begin{bmatrix} A_k^r & [0]_{5 \times 2} \\ [0]_{2 \times 5} & A^h \end{bmatrix} + \begin{bmatrix} B_k^r \\ [0]_{2 \times 2} \end{bmatrix} K_k, \tag{11b}$$

for $k \in \mathbb{N}_{[0,N-1]}$. The covariance propagation is given by

$$\Sigma_{k+1} = \Phi_k(x_k^r, u_k^r, \Sigma_k, K_k)$$
$$:= \check{A}_k \Sigma_k \check{A}_k^\top + \begin{bmatrix} [0]_{5\times2} \\ B^h \end{bmatrix} W_k^h \begin{bmatrix} [0]_{5\times2} \\ B^h \end{bmatrix}^\top, \quad (12)$$

for $k \in \mathbb{N}_{[0,N-1]}$. Note that the covariance propagation of the human state uncertainty, i.e.,

$$\Sigma_{k+1}^h = A^h \Sigma_k^h \{A^h\}^\top + B^h W_k^h \{B^h\}^\top,$$

depends on neither the robot nominal trajectory nor the feedback gain matrices. Therefore, the uncertain human trajectory can be precomputed and passed to the OCP as parameters. The robot state uncertainty Σ_k^r and the robot-human covariance Σ_k^{rh} are propagated inside the OCP:

$$\Sigma_{k+1}^r = \Phi_k^r(x_k^r, u_k^r, \Sigma_k, K_k) \quad (13a)$$
$$:= \begin{bmatrix} I_5 & [0]_{5\times2} \end{bmatrix} \Phi_k(x_k^r, u_k^r, \Sigma_k, K_k) \begin{bmatrix} I_5 & [0]_{5\times2} \end{bmatrix}^\top,$$
$$\Sigma_{k+1}^{rh} = \Phi_k^{rh}(x_k^r, u_k^r, \Sigma_k, K_k) \quad (13b)$$
$$:= \begin{bmatrix} I_5 & [0]_{5\times2} \end{bmatrix} \Phi_k(x_k^r, u_k^r, \Sigma_k, K_k) \begin{bmatrix} [0]_{2\times5} & I_2 \end{bmatrix}^\top,$$

for $k \in \mathbb{N}_{[0,N-1]}$.

3.2 Expected Stage Cost and Terminal Cost

The state and input uncertainties due to feedback policies affect the expected cost we aim to minimize. Considering the Gaussian distribution of the state and the linear state feedback policy (10), we have

$$\text{Cov}(\tilde{x}_k^r) = \Sigma_k^r \text{ and } \text{Cov}(\tilde{u}_k^r) = K_k \Sigma_k K_k^\top. \quad (14)$$

Given the trajectory-tracking cost defined in (9), the expected value of the stage cost is

$$\mathbb{E}\{l_k(\tilde{x}_k^r, \tilde{u}_k^r)\} = \text{Tr}\left\{\frac{1}{2}Q\Sigma_k^r\right\} + \text{Tr}\left\{\frac{1}{2}RK_k\Sigma_k K_k^\top\right\}$$
$$+ l_k(x_k^r, u_k^r) =: \tilde{l}_k(x_k^r, u_k^r, \Sigma_k, K_k).$$

The expected value of the terminal cost $\tilde{l}_N(x_N^r, \Sigma_N)$ can be defined similarly. We refer to (Messerer and Diehl, 2021) for the derivation.

3.3 Penalized Chance Constraints

Consider the stage constraints on the system state and input: $h_k(\tilde{x}_k, \tilde{u}_k) \leq 0 : \mathbb{R}^7 \times \mathbb{R}^2 \to \mathbb{R}^{n_{h,k}}$. Based on the linearization of the constraints on the nominal trajectory (x_k, u_k), the variance of one constraint component $h_{k,i}$ can be approximated by

$$\beta_{k,i} = H_{k,i}(x_k, u_k, \Sigma_k, K_k)$$
$$:= \nabla h_{k,i}(x_k, u_k)^\top \begin{bmatrix} I_7 \\ K_k \end{bmatrix} \Sigma_k \begin{bmatrix} I_7 \\ K_k \end{bmatrix}^\top \nabla h_{k,i}(x_k, u_k).$$

The constraint variance β_N for terminal constraints can be defined similarly. Due to the unbounded support of the Gaussian distribution, constraints can only be satisfied in a probabilistic manner (Schwarm and Nikolaou, 1999):

$$h_{k,i}(x_k, u_k) + \gamma\sqrt{\beta_{k,i}} \leq 0, \quad (15)$$

where γ is the number of standard deviations that the nominal trajectory should keep from the nominal bounds. Large disturbances can occasionally lead to infeasibility of state-dependent constraints of the OCP. Soft constraints are therefore recommended in practice:

$$\begin{cases} s_{k,i} \geq h_{k,i}(x_k, u_k) + \gamma\sqrt{\beta_{k,i}}, \\ s_{k,i} \geq 0, \end{cases} \quad (16)$$

where $s_{k,i}$ are slack variables. Weighted l_1 penalties $\tau_{k,i}s_{k,i}$ are added to the objective function. If the weights $\tau_{k,i}$ are

sufficiently large and the problem is feasible, the solution of the softened problem corresponds to a solution of the problem with hard constraints (Han and Mangasarian, 1979). In the infeasible case, the soft-constrained formulation will return a solution that minimizes constraint violation.

Remark 2. Incorporating feedback on human motion uncertainty can reduce the uncertainty of the relative position but meanwhile increases the uncertainty of the robot state. Therefore, the uncertainty of the relative position should only be reduced when necessary, i.e., when the collision avoidance constraints are active. The expected stage cost and terminal cost (15) encourage smaller uncertainty in the robot state and robot input in general.

Terminal Velocity Constraint It is desirable that the robot does in fact stand still at the end of the prediction horizon despite the uncertainty. Instead of imposing the penalized chance constraint (16) for the terminal forward velocity, we impose the nominal velocity constraint (7) and additionally restrict the variance of the forward velocity in the terminal state Σ_N^v to be less than a small ϵ_Σ value:

$$\Sigma_N^v \leq \epsilon_\Sigma \ll 1. \quad (17)$$

3.4 OCP Formulation

The covariance matrices Σ_k may not remain positive (semi)definite throughout optimization iterations when we treat the matrices Σ_k^r and Σ_k^{rh} as optimization variables in the OCP formulation. This results in potentially negative values for the constraint uncertainties β_k and issues when computing the square root (16). The problem can be resolved by initializing β_k with positive values and imposing

$$\beta_k \geq \epsilon_\beta \mathbb{1}_{n_{h,k}} \text{ and } \beta_k \geq H_k(x_k, u_k, \Sigma_k, K_k)$$

instead of $\beta_k = H_k(x_k, u_k, \Sigma_k, K_k)$ with ϵ_β being a small positive value (Messerer et al., 2023).

Given the assumption that the human does not respond to the robot motion, the human nominal trajectory can be precomputed. For compactness of notation, we define

$$Y_k^r := (x_0^r, \cdots, x_k^r, u_0^r, \cdots, u_k^r, K_0, \cdots, K_k), \text{ for } k \in \mathbb{N}_{[0,N-1]}.$$

The stochastic OCP incorporating full-state feedback can be formulated as

$$\min_{\substack{Y_{N-1}^r, \Sigma_0^r, \cdots, \Sigma_N^r, \\ \Sigma_0^{rh}, \cdots, \Sigma_N^{rh}, x_N^r, \\ s_0, \cdots s_N, \beta_0, \cdots, \beta_N}} \begin{array}{l} \sum_{k=0}^{N-1} \tilde{l}_k(x_k^r, u_k^r, \Sigma_k, K_k) \\ + \tilde{l}_N(x_N^r, \Sigma_N) + \sum_{k=0}^N \tau_k^\top s_k \end{array} \quad (18a)$$

$$\text{s.t.} \quad x_0^r = \bar{x}_0^r, \ x_{k+1}^r = \psi_k^r(x_k^r, u_k^r), \quad (18b)$$
$$\Sigma_0^r = [0]_{5\times5}, \ \Sigma_0^{rh} = [0]_{5\times2}, \quad (18c)$$
$$K_0 = [0]_{2\times7}, \quad (18d)$$
$$\Sigma_{k+1}^r = \Phi_k^r(x_k^r, u_k^r, \Sigma_k, K_k), \quad (18e)$$
$$\Sigma_{k+1}^{rh} = \Phi_k^{rh}(x_k^r, u_k^r, \Sigma_k, K_k), \quad (18f)$$
$$\beta_k \geq \epsilon_\beta \mathbb{1}_{n_{h,k}}, \ \beta_N \geq \epsilon_\beta \mathbb{1}_{n_{h,N}}, \quad (18g)$$
$$\beta_k \geq H_k(x_k, u_k, \Sigma_k, K_k), \quad (18h)$$
$$\beta_N \geq H_N(x_N, \Sigma_N), \quad (18i)$$
$$s_k \geq 0, \ s_N \geq 0, \quad (18j)$$
$$s_k \geq h_k(x_k, u_k) + \gamma\sqrt{\beta_k}, \quad (18k)$$
$$s_N \geq h_N(x_N) + \gamma\sqrt{\beta_N}, \quad (18l)$$
$$\epsilon_v \geq v_N^r \geq 0, \ \epsilon_\Sigma \geq \Sigma_N^v, \quad (18m)$$
$$\text{with } k \in \mathbb{N}_{[0,N-1]},$$

where $\bar{x}_0^{\mathrm{r}}$ is the value of the current robot state. The value of the current state uncertainty is set to zero because the current state is assumed to be known exactly. The terminal velocity constraint is written separately in (18m) and not included in (18l).

3.5 Feedback Policy Variations

The linear feedback policy defined in (10) corresponds to incorporating feedback on both the robot state and the human state. It will be referred to as *full*-state feedback for the remainder of this paper. Apart from the full-state feedback policy, another option is to provide feedback solely on the human motion uncertainty, i.e., $K_k^{\mathrm{h}}\left(\tilde{x}_k^{\mathrm{h}}-x_k^{\mathrm{h}}\right)$, with $K_k^{\mathrm{h}} \in \mathbb{R}^{2 \times 2}$ being the feedback gain for the human state deviation. The feedback law introduces uncertainty to the robot velocity, due to which constraint (17) can be difficult to satisfy once the OCP formulation models feedback on the human state deviation. Therefore, the robot needs to be able to reduce its own forward velocity uncertainty:

$$\tilde{u}_k^{\mathrm{r\text{-}PRTL}}(\tilde{x}_k) := u_k^{\mathrm{r}} + K_k^{\mathrm{h}}\left(\tilde{x}_k^{\mathrm{h}} - x_k^{\mathrm{h}}\right) + K_k^{\mathrm{r},v}\left(\tilde{v}_k^{\mathrm{r}} - v_k^{\mathrm{r}}\right), \quad (19)$$

for $k \in \mathbb{N}_{[1,N-1]}$, where $K_k^{\mathrm{r},v}$ is a scalar. The feedback policy defined in (19) will be referred to as *partial*-state feedback.

The capability of feeding back the robot forward-velocity uncertainty is particularly essential towards the end of the prediction horizon. Therefore, it can be an implementation choice to allow non-zero $K_k^{\mathrm{r},v}$ for only a subset of time instants. The OCP formulation corresponding to the partial-state feedback includes only the non-zero feedback gain elements, i.e., K_k^{h} and $K_k^{\mathrm{r},v}$, in the optimization variables.

Remark 3. Compared with the partial-state feedback policy, treating the full feedback gain matrices as decision variables incurs an increase of optimization variables and the nonlinearity in the optimization problem. On the other hand, it enables the robot to counteract its state uncertainty gained from the feedback on the human motion uncertainty. Moreover, it can handle more general situations where the robot motion itself is subject to process noise.

Incorporating feedback contrasts with sticking to nominal input trajectories irrespective of future realizations of disturbance. The former is referred to as *closed-loop* stochastic OCP while the latter as *open-loop* stochastic OCP for the remainder of this paper. The open-loop stochastic OCP can also be regarded as a special case where $K_k = [0]_{2 \times 7}$ for $k \in \mathbb{N}_{[1,N-1]}$.

4. RESULTS

We first present the OCP planned trajectories. We subsequently evaluate the MPC trajectories simulated based on different realizations of human motion uncertainty. The assessment of computation time is presented at the end. We implement the OCPs in CasADi (Andersson et al., 2019). We use the solver IPOPT (Wächter and Biegler, 2006) to solve the nonlinear programming problems with MA-27 (HSL, 2011) as the linear solver. The computations are carried out on a computer with an Intel(R) Xeon(R) W-1250P CPU and 32GB of RAM.

4.1 OCP Planned Trajectories

Test Scenario We simulate a robot driving in a corridor. The robot reference trajectory is traveling along the x-axis at a constant velocity:

$$x_k^{\mathrm{r,ref}} = \left[\ \left(\bar{p}_x^{\mathrm{r}} + v_x^{\mathrm{r,ref}} \cdot (t^{\mathrm{sim}} + k\delta_t)\right)\ 0\ 0\ v_x^{\mathrm{r,ref}}\ 0\ \right]^\top,$$

Table 1. Parameters used in simulation in Sec. 4

Name	Unit	Symbol	Value
Discretization intervals		N	20
Safety distance	m	Δ_{safe}	0.3
Human velocity covariance	m²·s⁻²	W_k^{h}	$0.4^2 \cdot I_2$
Weight of position tracking error	m⁻²		50
Weight of velocity tracking error	s²·m⁻²		2
Weight of input tracking error	s⁴·m⁻²		2

for $k \in \mathbb{N}_{[0,N]}$, where $\bar{p}_x^{\mathrm{r}}$ is the robot x-axis position at the beginning of the simulation, t^{sim} is the time in simulation, and $v_x^{\mathrm{r,ref}}$ is the reference velocity. A human walks in the corridor in the opposite direction and the nominal velocity is constant: $x_k^{\mathrm{h}} = \left[\ \left(\bar{p}_x^{\mathrm{h}} + v_x^{\mathrm{h}} \cdot (t^{\mathrm{sim}} + k\delta_t)\right)\ \bar{p}_y^{\mathrm{h}}\ \right]^\top$, for $k \in \mathbb{N}_{[0,N]}$, where $\left[\ \bar{p}_x^{\mathrm{h}}\ \bar{p}_y^{\mathrm{h}}\ \right]^\top$ is the human state at the start of the simulation. Although the nominal human velocity u_k^{h} is constant in this test scenario, the velocity u_k^{h} can be time-varying. The determination of the values of u_k^{h} alongside the uncertainty matrix W_k^{h} is not the focus of this paper. The robot state and the human state are exactly known at the initial state of the OCP. The OCP parameters are collected in Table 1.

Examples of the OCP planned trajectories for the number of standard deviations γ equal to three are shown in Fig. 1 and Fig. 2. Note that the overlap between the orange human ellipsoids and the blue semi-transparent sets does *not* imply that the collision chance constraints are violated. The overlap is due to the correlation between the robot state uncertainty and human state uncertainty arising from the feedback scheme. The correlation indicates that the robot will respond to the human motion uncertainty.

When the robot meets the human at the end of the prediction horizon, the optimized feedback policy counteracts human motion uncertainty by adapting the robot angular acceleration, particularly the uncertainty in the direction perpendicular to the robot forward motion. It enables the robot to stay closer to the reference trajectory (see Fig. 1a). Meanwhile, the robot gains uncertainty, particularly in its angular velocity (see Fig. 2a).

In the scenario that the robot manages to pass the human in the prediction horizon, the robot applies feedback while passing by the human, especially via the forward acceleration. It results in uncertainty in the robot forward velocity, as highlighted by the dotted-line red ellipsoid in Fig. 2b. After the robot safely passes the human, the full-state feedback enables the robot to reduce its own state uncertainty through feedback. In consequence, the robot state turns out relatively certain at the end of the prediction horizon (see Fig. 1b). In contrast, in the case of partial-state feedback, the robot state, especially the robot position, is remarkably more uncertain (see Fig. 1c).

4.2 MPC Simulation

The human and robot trajectories running open-loop and closed-loop stochastic MPCs are shown in Fig 3. The robot steered by open-loop stochastic MPC (OL-SMPC) deviates more from the reference trajectory and is slower compared to closed-loop stochastic MPC (CL-SMPC). The difference of counteracting solely the human state uncertainty (P-CL-SMPC) versus both the human and the robot uncertainty (F-CL-SMPC) is comparatively small.

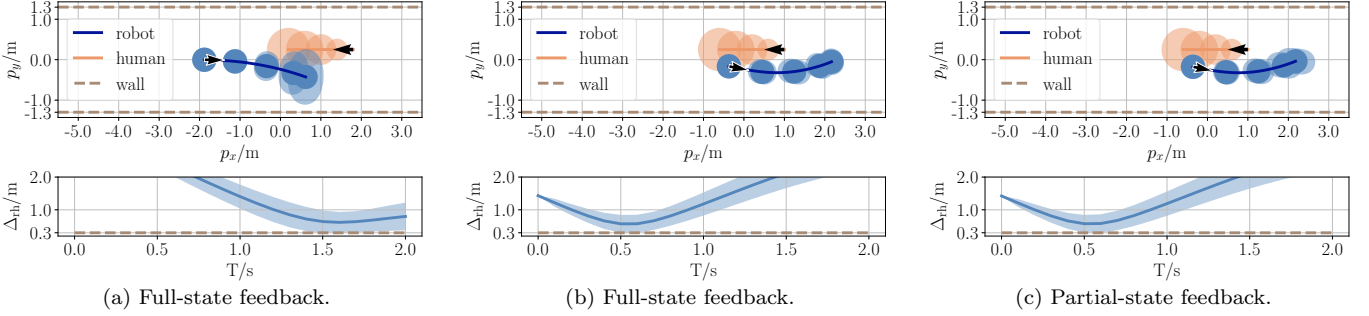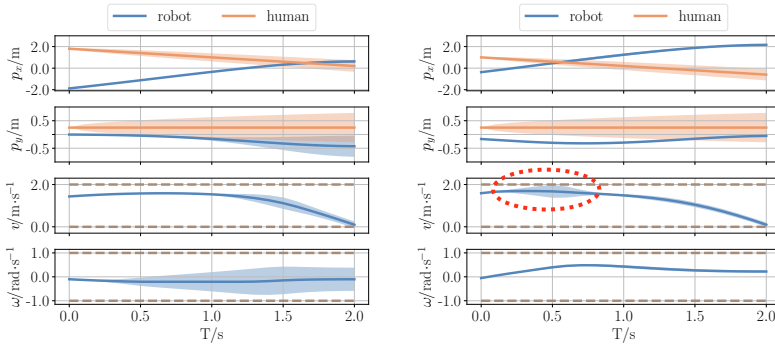(a) Full-state feedback.   (b) Full-state feedback.   (c) Partial-state feedback.

Fig. 1. Planned trajectories. The solid blue lines and solid orange lines in the upper plots represent the nominal trajectory of the robot and human respectively. The blue circles are of radius Δ_{safe}, and they correspond to time instant $0.0\,\text{s}$, $0.5\,\text{s}$, $1.0\,\text{s}$, $1.5\,\text{s}$, and $2.0\,\text{s}$ respectively. Each blue semi-transparent set depicts the Minkowski sum of a radius-Δ_{safe} circle and the ellipsoid corresponding to the three standard deviations of the robot position uncertainty. The orange ellipsoids and the shadows in the distance plots represent uncertainty tubes (three standard deviations).

(a) Robot and human meet in the later part of the prediction horizon (Fig. 1a). (b) Robot and the human pass by within the prediction horizon. (Fig. 1b).

Fig. 2. Planned robot state trajectories in time. Full-state feedback is incorporated. The shadows represent uncertainty tubes (three standard deviations). The red ellipsoid is for highlighting purpose.
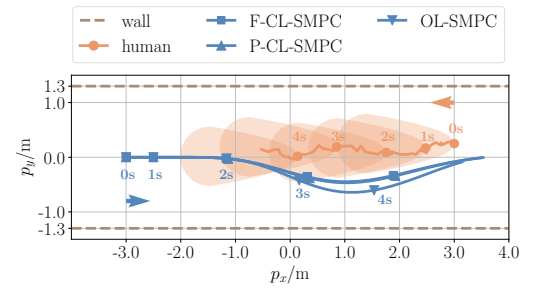

Fig. 3. Robot and human trajectories in an MPC simulation of five seconds. The blue markers depict the robot positions at certain time instants. The orange semi-transparent sets correspond to the modeled human occupancy in space (three standard deviations).

We test the collision likelihood and cost values given 1000 human trajectory samples for $\gamma = 3.0$ and $\gamma = 2.0$ respectively. The numerical solver occasionally fails to converge to a solution. In such cases, we use the control policies optimized in previous time steps. In each five-second long simulation, the robot and human meet once. One simulation is aborted once a collision takes place. For each simulation, we compute the mean of the trajectory-tracking stage cost (9a) evaluated for $k = 0$ at simulated time steps. The mean stage cost and the minimum robot-human distance of different simulations are shown in Fig. 4. The median of the mean stage cost and the number of collisions over 1000 simulations are reported in Table 2.

Nominal MPC neglects human motion uncertainty and therefore shows a high likelihood of collision. OL-SMPC attains low collision frequency while keeping an overly large distance to the human (see the markers in the dotted-line red ellipsoids in Fig. 4). This results in, on average, a relatively high trajectory-tracking cost (see Table. 2). The stage cost goes high when the robot gets stuck. Such cases are more frequent when the robot is steered by OL-SMPC compared with CL-SMPC (see Fig. 4). The tracking performance associated with P-CL-SMPC is similar to that of F-CL-SMPC, as measured by the trajectory-tracking stage cost.

4.3 Computation Time Evaluation

In addition to the straight-line corridor scenario, the robot is anticipated to navigate curved paths. The curvature of the reference trajectory potentially impacts the degree of nonlinearity involved in the stochastic OCP. Therefore, for the evaluation of computation time, we employ a modified test scenario: the robot navigates along a circular arc while the human participant travels along a concentric arc. Our experiment involves solving 300 OCPs, with variations in the starting and the goal positions of the robot as well as the curvature of the circular arcs.

The initial guess of the state variables for the nominal OCP is derived from the reference trajectory, for all other OCPs the initial guess is derived from the solution of the nominal

Table 2. Number of occurred collisions and median of the mean stage cost over 1000 simulations.

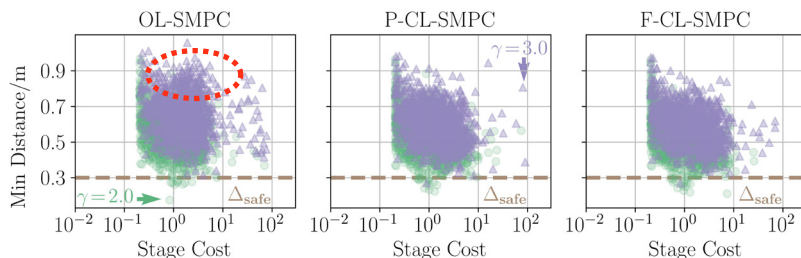|		# collisions	Stage cost (9a)
	Nominal MPC	305	0.25
$\gamma = 3.0$	OL-SMPC	0	1.66
	P-CL-SMPC	1	1.29
	F-CL-SMPC	1	1.27
$\gamma = 2.0$	OL-SMPC	10	0.98
	P-CL-SMPC	17	0.88
	F-CL-SMPC	18	0.84

Fig. 4. Mean stage cost and minimum robot-human distance. Purple triangles and green circles correspond to $\gamma = 3.0$ and $\gamma = 2.0$.
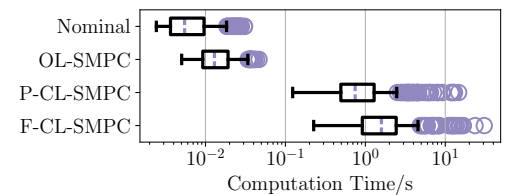

Fig. 5. Computation time of solving OCPs ($\gamma = 3.0$). The box extends from the lower to upper quartile of the data. The dashed line shows the median.

OCP. The computation time is reported in Fig. 5. For OL-SMPC, the covariance propagation can be precomputed. The covariance matrices are passed to the OCPs as optimization variables while the constraint uncertainties β_k of nonlinear constraints are optimized within the OCPs. The median computation time for nominal OCPs, OL-SMPC, P-CL-SMPC, and F-CL-SMPC is 5.5 milliseconds, 13 milliseconds, 0.75 seconds, and 1.6 seconds respectively.

Remark 4. There still exist cases that the numerical solver fails to converge although the OCP is feasible. Further improvement of numerical performance is subject to future research. One practical approach is to adopt a homotopy optimization scheme, where we start with solving a nominal OCP and gradually increase the scale of the uncertainty until we reach the scale of the modeled uncertainty. This approach improves the reliability of the solver at the cost of increasing the computation time. Although the computation time is relatively long, it is possible to use the optimized feedback policy for a few time steps while the new OCP is being solved.

5. CONCLUSION

This paper presents a stochastic-MPC-based method for robot motion planning which incorporates feedback on human motion uncertainty. The OCP planned trajectories and MPC trajectories of different feedback policies are compared in simulation. The partial feedback policy overall strikes a favorable balance between computational speed and performance. Future work could focus on modeling the human feedback on the robot motion and efficient algorithms to solve the closed-loop stochastic OCP.

REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – a software framework for nonlinear optimization and optimal control. *Math. Program. Comput.*, 11(1), 1–36.

Brock, O. and Khatib, O. (2002). Elastic strips: A framework for motion generation in human environments. *Int. J. Robotics Res.*, 21(12), 1031–1052.

Carvalho, A., Gao, Y., Lefevre, S., and Borrelli, F. (2014). Stochastic predictive control of autonomous vehicles in uncertain environments. In *Proc. 12th Int. Symp. Adv. Vehicle Control*.

Castillo-Lopez, M., Ludivig, P., Sajadi-Alamdari, S.A., Sanchez-Lopez, J.L., Olivares-Mendez, M.A., and Voos, H. (2020). A real-time approach for chance-constrained motion planning with dynamic obstacles. *IEEE Robotics and Automat. Lett.*, 5(2), 3620–3625.

Du Toit, N.E. and Burdick, J.W. (2012). Robot motion planning in dynamic, uncertain environments. *IEEE Trans. Robotics*, 28(1), 101–115.

Goulart, P.J., Kerrigan, E.C., and Maciejowski, J.M. (2006). Optimization over state feedback policies for robust control with constraints. *Automatica*, 42, 523–533.

Han, S.P. and Mangasarian, O.L. (1979). Exact penalty functions in nonlinear programming. *Math. Program.*, 17(1), 251–269.

HSL (2011). A collection of Fortran codes for large scale scientific computation. http://www.hsl.rl.ac.uk.

Macek, K., Vasquez, D., Fraichard, T., and Siegwart, R. (2008). Towards safe vehicle navigation in dynamic urban scenarios. *Automatika*, 50, 184–194.

Mayne, D.Q., Kerrigan, E.C., and Falugi, P. (2011). Robust model predictive control: advantages and disadvantages of tube-based methods. *Proc. IFAC World Congr.*, 44(1), 191–196.

Messerer, F., Baumgärtner, K., and Diehl, M. (2023). A dual-control effect preserving formulation for nonlinear output-feedback stochastic model predictive control with constraints. *IEEE Contr. Syst. Lett.*, 7, 1171–1176.

Messerer, F. and Diehl, M. (2021). An efficient algorithm for tube-based robust nonlinear optimal control with optimal linear feedback. In *Proc. IEEE Conf. Decis. Control (CDC)*.

Nagy, Z. and Braatz, R. (2004). Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. *J. Process Control*, 14, 411–422.

Schwarm, A. and Nikolaou, M. (1999). Chance-constrained model predictive control. *AIChE J.*, 45(8), 1743–1752.

Schwarting, W., Alonso-Mora, J., Paull, L., Karaman, S., and Rus, D. (2018). Safe nonlinear trajectory generation for parallel autonomy with a dynamic vehicle model. *IEEE Trans. Intell. Transp. Syst.*, 19(9), 2994–3008.

Scokaert, P.O.M. and Mayne, D.Q. (1998). Min-max feedback model predictive control for constrained linear systems. *IEEE Trans. Automat. Contr.*, 43, 1136–1142.

Siegwart, R., Nourbakhsh, I.R., and Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*. MIT Press, Cambridge, MA, USA.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1), 25–57.

Yan, J. and Bitmead, R. (2005). Incorporating state estimation into model predictive control and its application to networktraffic control. *Automatica*, 41, 595–604.

Zhu, H. and Alonso-Mora, J. (2019). Chance-constrained collision avoidance for MAVs in dynamic environments. *IEEE Robotics and Automat. Lett.*, 4(2), 776–783.